TS EAS

Thursday 6th of August

9.00 pm Wednesday PDT / 10.00 pm Wednesday MDT / 11.00 pm CDT / 12.00 am Wednesday/Thursday EDT / 5.00am BST / 6.00 am CEST/ 2.00pm AEST

biogHist

Kerstin

Current status

- Difference in terms of spelling solved by general spelling vote taken by TS-EAS:
  - Will be moving to camelCase spellings for both EAS

- Differences in terms of attributes solved via the general decisions on attributes usage
  - EAC-CPF will include:
    - @id, @languageOfElement (renamed), @scriptOfElement (added for reconciliation), @localType (?), audience (added for reconciliation)
  - EAD3 will include:
    - @id, @languageOfElement (renamed), @scriptOfElement (renamed), @localType (?), audience
    - To be reviewed: @altrender, @encodinganalog

Current status

- Main differences still exist in terms of sub-elements:
  - More detailed overview in the document named “2020-08-06_Topic_BiogHist_TS_EAS_Meeting.pdf” (especially for EAC, EAD and Schema teams)
  - Three groups of elements to talk about:
    - Those available and used in both EAS
    - Those used only in one EAS, though generally available in the other
    - Those used only in one EAS and (currently) not available in the other

Pending questions

- <list> and <p>
  - Find technical solution via schema definition to deal with differences in terms of additional formatting and mixed content element
  - Review option to use <span@localType> as a minimal approach to mixed content in upcoming major revision of EAD?
- <chronList> and its sub-element <chronItem>
  - Allow for all three date elements in <chronItem> in EAC-CPF?
  - Enable <chronItemSet> in EAC-CPF to pair a date with one or more events and zero or more geographic names?

Pending questions

- <citation>
  - Decision on Tuesday to replace <citation> with <ref>, which will be renamed to <reference>
  - Use <reference> as sub-element of <p> rather than <biogHist> directly?
- <abstract>
  - Review using <abstract> as a dedicated summary element with <biogHist> and potentially other informal descriptive elements in EAD as part of the major revision?
- <biogHist>
  - What are the use cases for nesting <biogHist> (and other informal descriptive elements) within itself?

Pending questions

- <blockquote>, <head>, and <table>
  - Find technical solution via schema definition to deal with differences in terms of additional formatting
- <outline>
  - What are the use cases for <outline>? How useful is this element in general?

EAC Roundup / Test Corpus

Mark

Files gathered so far

- Actorenregister Nationaal Archief: 5,800
- American Museum of Natural History: 8
- American Numismatics Society: 13
- Bibliothèque Nationale de France: 101
- Connecting the Dots: 80
- Social Networks and Archival Context: 8,643 (and counting)
- Turnbull Library: 200,866

A few stats from 215,459 files

- relationEntry 1,575,434
- persname 1,378,251
- resourceRelation 1,044,410
- p 972,695
- descriptiveNote 747,022
- cpfRelation 531,024
- **objectXMLWrap 401,713**

- nameEntry 345,370
- language 343,801
- abstract 329,216
- physdesc 329,116
- extent 329,116
- unittitle 329,116
- did 329,116

A few stats from 215,459 files

- languageDeclaration 212,023
- biogHist 211,326
- existDates 165,169
- dateRange 132,410
- fromDate 123,968
- toDate 108,331

- date 62,048
- localDescription 57,656
- placeEntry 56,091
- place 52,018
- citation 40,310
- eventDescription 35,041

A few stats from 215,459 files

- conventionDeclaration 14857
- abbreviation 6285
- otherRecordId 5919
- entityId 5901
- localControl 5901
- nameEntryParallel 3433
- functions 3328
- snac:placeEntry 2566
- span 1924
- localTypeDeclaration 116
- outline 4
- multipleIdentities 1

Regarding attributes:

- @localType 138,212
- @snac:preferenceScore 35,235
- @accuracy 0
- @latitude 0
- **@xml:base** 0

objectBinWrap

Mark

<objectBinWrap> **Object Bin Wrap** [toc]

Summary: This element provides a place for a base64-encoded binary representation of a resource.

May contain: base64Binary

May occur within: cpfRelation, functionRelation, resourceRelation, setComponent, source

Description and Usage: This element provides a place for a base64-encoded binary representation of a resource. The datatype of base64-encoded binary is based on the W3C Schema Part 2: Datatypes. (for which consult the specification at http://www.w3.org/TR/xmlschema-2/)

Attributes:

xml:id	Optional

Availability: Optional, Non-repeatable

Example:

```
<objectBinWrap> [Base64 Binary code] </objectBinWrap>
```

Given that...

- We have no example in the Tag Library;
- No known usage of the element;
- This element was *not* added to EAD3;
- The “May contain” base64binary link in the tag library is broken (shouldn’t be a link);
- We already decided to rename the <script> element due to security concerns mentions via this issue in the EAD3 repository: https://github.com/SAA-SDT/EAD3/issues/520;
- EAC-CPF and EAD are not (and need not be) METS;
- And, again, no known usage in EAC-CPF...

I propose that we...

- Remove <objectBinWrap> from the EAC-CPF

Any objections? Suggestions? etc.

objectXMLWrap

Mark

<objectXMLWrap> **Object XML Wrap** [toc]

Summary: A place for incorporating XML elements from any XML namespace.

May contain: any element from any namespace

May occur within: cpfRelation, functionRelation, resourceRelation, setComponent, source

Description and Usage: This element provides a place to express data in another XML encoding language. While the element is not restricted with respect to namespace, to facilitate interoperability, the XML should conform to an open, standard XML schema and a namespace attribute should be present on the root element referencing the namespace of the standard.

Attributes:

xml:id	Optional

Availability: Optional, Non-repeatable

Examples:

```
<objectXMLWrap>
    <mods xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.loc.gov/mods/v3 http://www.loc.gov/mods/v3/mods-3-
    3.xsd">
        <titleInfo>
            <title>Artisti trentini tra le due guerre</title>
        </titleInfo>
        <name>
            <namePart type="given">Nicoletta</namePart>
            <namePart type="family">Boschiero</namePart>
            <role>
                <roleTerm type="text">autore</roleTerm>
            </role>
        </name>
    </mods>
</objectXMLWrap>
```

Questions

- Do we want to be METS-like?
- If we keep objectXMLWrap, should it allow elements in any namespace, including the current one?
- If we keep it, why not a shorter name like xmlData?
- This would require research, but what if we had a way for “power users” to extend the schema when needed to add whatever XML they needed (and in more places, like biogHist)? NVDL is a great candidate for this. But, the question would be: would folks who are not power users benefit from the generic approach afforded by objectXMLWrap.
    - TEI
    - HTML
    - ITS: https://www.w3.org/TR/its20/ (which is extremely important if we really want to support internationalization)

@localType

Karin

Background

- In Berlin we came to: Could be part of the more general conversation about @localType vs context-specific @...type attributes vs simply @type.
- @localType was introduced in EAC-CPF for being able to give the users own value lists to elements. When EAD3 was created it was moved over and @type as previously used in EAD 2002 was changed to having more distinct names when value lists were provided as part of the schema definition, @localType was added for giving own values and thus @type was removed.

Possible solutions 1(2)

- Keep as is with no changes, i.e. using only @localType with the implementer’s values in EAC-CPF and using @localtype plus context-specific @...type-s in EAD.
- Rename context-specific @...type-s to simply "type" when it's using predefined values from the schema and rely on the context to clarify what this "type" refers to.
    - Keep @localType for the user’s own values; or
    - Consider an approach with having the value "other" for @type, followed by an @othertype attribute with open values instead of @localType.
    - Would imply according name change to <localTypeDeclaration>.

Possible solutions 2(2)

- Review use cases for @localType and consider using other attributes that would fulfil the same purpose.
  - E.g. @vocabularySource from the new <relations> proposal to indicate the originating system for <otherRecordId>-s, which is currently given with @localType in some cases.
  - Review other attributes and contexts, where something else than @localType could be defined as using the implementer's own values in some way.
  - Would imply according name change to <localTypeDeclaration>.
- Introduce the option for users to add their own attributes to all elements, thereby eliminating the need for a @localType to be made available in the EAS directly.
- Other directions?

Discussion points

- The solutions suggested
- How does adding xs:anyattribute affect interoperability and sense and purpose of EAD/EAC as an exchange format?
- Will adding the three attributes (@vocabularySource, @vocabularySourceURI, and @valueURI) give wiggle room in other contexts than the mentioned example of <otherRecordId> so that implementers can just use one of them without having a vocabulary so it is in fact the @???type ???
- Have the addition of the target referencing attributes in control that will be added made a change to this question?

Comments given

- Mike Rush: Whatever the group chooses, just don't introduce a generically named @type attribute. (https://github.com/SAA-SDT/eac-cpf-schema/issues/15#issuecomment-378241463)
- Silke: Keep as is:
  - Use @localType as generic type attribute to all (?) elements in //cpfDescription and //relation.
  - Add specific type attributes, if needed, eg where a defined value list is provided.
  - Revise the usage of @localType within //control.
  - Add this practise to EAS schema design principles.
- Mark:
  - In the recordID example the use of the 3 new attributes makes it not needed with a @localType.
  - Allow/add xsd:anyAttribute. We only get one localType, and folks might want to add multiple attributes...
  - The suggestion from the Design Principles is not to allow any attribute whatsoever, but instead to allow any attribute in a namespace. I don't see how this affects interoperability at all. The foreign namespaced attributes, if any, would be safely ignored.

What do we do?

We need to make our mind up before the EAC-CPF RFC is opened.

See you tomorrow!

Friday 7th of August
6.00 am PDT / 7.00 am MDT / 8.00 am CDT / 9.00 am EDT / 2.00pm BST /
3.00 pm CEST / 11.00pm AEST